COS 4.0

Differences from COS 3.4

for

380Z Varitext (80/40 character) systems

RESEARCH
MACHINES

Reference Notes

Differences between COS 3.4 and COS 4.0

Reference Notes

Copyright (c) 1982 by Research Machines Limited

All rights reserved. Copies of this publication may be made by customers exclusively for their own use, but otherwise no part of it may be reproduced, transmitted, transcribed, stored in a retrieval system, or translated into any language or computer laguage without the prior written permission of Research Machines Limited, Post Office Box 75, Oxford, England. OX2 0BW Tel: Oxford (0865) 249866.

The policy of Research Machines Limited is one of continuous development and improvement of its products and services, and the right is therefore reserved to revise this document or to make changes in the computer software it describes without notice. Research Machines Limited endeavour to ensure the accuracy of this document and to ensure that the products described perform correctly according to their descriptions. However, Research Machines Limited do not accept liability for the consequences of any error or omission.

Additional copies of this publication may be ordered from Research Machines at the address above. Please ask for 'Differences between COS 3.4 and COS 4.0 - Reference Notes'.

Research Machines Limited is committed to the policy of continuous improvement of its products and services. To assist the Company to improve the quality of its documentation and computer software it would like to receive reports from users on their experience with using them.

Reports referring to documentation should include:

- a description of any errors you have found;
- comments on problems you have encountered with understanding and using the manual, and suggestions for improvement;
- your name and address and a brief description of your experience with using microcomputer systems.

Queries on documentation should be sent to:

The Documentation Department,
Research Machines Limited,
PO Box 75,
Oxford.
OX2 0BW

Reports referring to software errors should include:

- a written description of each step taken and each key or command you type.
- details of the hardware configuration, and the version numbers of the software in use.

Queries on software errors should be sent to:

Technical Support Department,
Research Machines Limited,
PO Box 75,
Oxford.
OX2 0BW

CONTENTS

SECTION 1:	INTRODUCTION	1.1
SECTION 2:	NEW FEATURES	2.1
	Escape Sequences	2.1
	Screen Width	2.1
	Smooth Scrolling	2.1
	Downward Scrolling	2.2
	Scrolling Windows	2.2
	Attributes	2.3
	Writable Characters	2.3
	Special Characters	2.4
	Initialisation	2.4
	Clearing area of screen	2.4
	Beeper	2.5
SECTION 3:	CHANGES TO FRONT PANEL AND COS COMMANDS	3.1
	Front Panel	3.1
	COS Commands	3.1
SECTION 4:	EMT DIFFERENCES	4.1
	Modified EMTs	4.1
	Specification of new EMTs	4.3
SECTION 5:	ARCHITECTURE DIFFERENCES	5.1
	Memory maps	5.1
	Addresses used by COS	5.1
	Screen access methods	5.2

APPENDICES

Escape Sequences
Effect of control characters on the screen
COS 4.0 Quick Reference Guide

 Copyright January 1982

1. INTRODUCTION

This document describes the principal differences between COS 4.0, the firmware of a 380Z Varitext (80/40 character) system, and COS 3.4, the firmware of a 380Z 40-character system. It is intended to be read in conjunction with a COS 3.4 Reference Manual.

New screen features include:-

- 80-character mode
- Smooth Scrolling
- Downward scrolling
- Scrolling windows
- Character attributes, including dim, reverse video, and underlined
- A writable character store
- Support for a 'beeper'

These features are accessible from higher level languages by outputting special 'escape sequences' of characters to the screen, using the PRINT or PUT statements in BASIC for example. They are also available to the assembler programmer via the EMT mechanism.

Some features of COS 3.4 are not supported, e.g. cassette support other features are only available unchanged when the screen is in 40-character mode.

Section 2 describes the new features provided by COS 4.0 and how to use them from BASIC.

Section 3 describes the changes to the Front Panel and to COS command level.

Section 4 describes differences in the EMTs available, including the new ones which give access to the new screen handling features.

Section 5 describes other ways in which the system architecture differs from that of a 40-character system.

Appendices define the escape sequences available, and the effect of control characters output to the screen, for reference purposes.

 Copyright January 1982

2. NEW FEATURES

2.1 Escape Sequences

Before describing the new features it is necessary to explain the concept of an 'escape sequence', which is the means by which most of the features are made available to programmers who use high level languages.

An escape sequence is a sequence of characters output to the screen handler. The sequence consists of an ESC character (ASCII 27), followed by a sequence of characters which define what action is required. None of the characters which form an escape sequence appears on the screen.

Typing the characters on the keyboard is only an effective way of entering an escape sequence if the characters are all echoed to the screen. Most software does not echo the escape character.

2.2 Screen Width

The screen width can be switched into 40-character mode by the escape sequence

ESC, '=', '0', 'J'

and to 80-character mode by

ESC, '=', '1', 'J'

For example, from BASIC, the screen can be forced into 40-character mode by

```
PUT 27, "=0J"
```

2.3 Smooth Scrolling (80-character mode only).

It is possible to go into a mode called smooth scrolling, which means that all scrolling is slower than usual, and less jerky. There is no other difference.

Smooth scrolling may be initiated or terminated from the keyboard by typing CTRL @ which acts in a similar way to that in which typing CTRL A enables and disables autopaging.

Smooth scrolling may also be enabled by outputting the escape sequence

ESC, '=', '1', 'H'

to the screen, and may be disabled by the sequence

ESC, '=', '0', 'H'

For example, to enable smooth scrolling from a BASIC program, use the statement

```
PUT 27, "=1H"
```

 Copyright January 1982

2.4 Downward scrolling

If a CTRL K (ASCII 11) is sent to the screen, the cursor will move up one line. If the cursor is already at the top of the screen, then sending CTRL K will cause the screen to scroll down one line. Please note that it is not possible to smooth scroll downwards, so even if smooth scrolling is active, scrolling downwards will be a normal scroll.

So, for example, the following BASIC program will print the numbers 1 to 100 upwards from the bottom of the screen.

```
10 FOR I= 1 TO 100
20 PRINT I;CHR$(14);CHR$(11);
30 NEXT I
```

Executing the BASIC statement

```
NULL 2,11
```

will have the effect of making BASIC scroll downwards at all times. (The effect can be undone with the statement NULL 0).

2.5 Scrolling Windows

It is possible to define any rectangular area of the screen as the scrolling window. This would mean that any character which was printed on the screen would only appear inside the window, and only the contents of the window would be scrolled.

A window may be defined using the escape sequence

```
ESC, '?', xl, xu, yl, yu, 'A'
```

where `0 <= xl <= xu <= 79` (39 in 40-character mode)
`0 <= yl <= yu <= 23`

This results in columns xl to xu on rows yl to yu inclusive being defined as the new scrolling window.

For example, in BASIC,

```
PUT 27, "?", 0, 39, 12, 23, "A"
```

would define as a scrolling window the bottom left hand quarter of the screen in 80-character mode, or the bottom half of the screen in 40-character mode. Immediately after defining a window, the cursor will go to the bottom left hand corner of the new window. To revert to the normal full screen scrolling, define that as the window, or, in the case of BASIC, execute the TEXT command. It is possible to scroll downwards in windows. It is not possible to smooth scroll in windows, so even if smooth scrolling is active, all scrolling in a window will be normal scrolling.

Assembler programmers can also use EMT WINDOW.

2.6 Attributes

Characters printed to the screen can have any combination of the following attributes:-

Dim
Reverse video
Underlined

Attributes can be switched on and off by escape sequences of the form

ESC, '=', n, c

where n = '0' to switch an attribute off
n = '1' to switch an attribute on
c = 'B' for underline
c = 'C' for dim
c = 'D' for reverse video

For example, to print 'HELLO' in underlined dim characters the following BASIC statements could be used.

```
PUT 27, "=1B", 27, "=1C", "HELLO"
```

The attributes remain in force until turned off. To return to normal, the attributes could be turned off individually or the escape sequence

ESC, '>', 'C'

could be used to reset all attributes to normal.

Reverse video can also be switched on by CTRL R and off by CTRL T.

An additional attribute, 'alternate character set', is provided for, but no such set is implemented in the current system. An attempt to use it will give undefined results.

Assembler programmers can also use EMT VTOUT to output characters with attributes.

2.7 Writable Characters

The characters from ASCII 128 to 255 are normally the low resolution graphics characters. However, some or all of them can be redefined. To define a character requires 12 bytes. A character is 8 dots wide by 10 deep. The first 10 bytes correspond to the 10 lines of dots. Every bit which is set will give a dot in the corresponding position. The 11th and 12th bytes are the underline, and replace lines 10 and 9 when the underline attribute is on. To define a character, use the escape sequence:-

ESC, '<', C, b1, b2, b3, b4, b5, b6, b7, b8, b9, b10, b11, b12

where c is the code of the character to be defined and b1 to b12 are the bytes to define the character. For example, character 192 can be redefined as the Greek character rho and output by

```
PUT 27, "<", 192, 0, 0, 0, 0, 24, 36, 68, 120, 64, 64, 64, 255
PUT 192
```

It should be noted that characters which are in range ASCII 128 to 191 automatically appear dim on the screen, by hardware in 40 character mode and by software in 80 character mode. It is possible to switch this feature on or off in 80 character mode. Note that if automatic dimming is switched off, then graphics which are normally dim will now be bright. The escape sequence which switches dimming off is

```
ESC, '=', '0', 'K'
```

The Writable Character store can be reinitialised to contain the usual low resolution graphics characters by the escape sequence

```
ESC, '>', 'B'
```

Assembler programmers can also use EMT CHGEN to define characters.

2.8 Special Characters

Characters from ASCII 0 to 31, and also 127 have special effects when output to the screen. However, these codes have defined character representations associated with them. (See, for example, the 380Z Disc System Information File.) Such a character can be written to the screen using the escape sequence

```
ESC, '!', c
```

where c is the desired code.

For example, to output the symbol pi to the screen from BASIC:-

```
PUT 27, "!", 21
```

2.9 Initialisation

A standard initial state can be reached using the escape sequence

```
ESC, '>', 'A'
```

which clears attributes, clear smooth scrolling, sets 80-character mode and full screen window. It also re-initialises the Writable Character Store.

2.10 Clearing area of screen

A rectangular area of screen can be cleared using the escape sequence

```
ESC, '?', xl, xu, yl, yu, 'B'
```

where
```
0 <= xl <= xu <= 79      (39 in 40-character mode)
0 <= yl <= yu <= 23
```

This clears the rectangle whose opposite corners are (xl, yl) and (xu, yu).

Assembler programmers can use EMT VTCLR.

2.11 Beeper

There is a facility for attaching a loudspeaker to the 380Z. A single note 'beep' is sounded when CTRL G is output to the screen. (Note that the standard hardware does not include a speaker, but that one can be attached between pins 6 (beeper) and 7 (ground) of the 7-way DIN Socket marked 'Cassette Interface' on the rear panel. NEC monitors have a suitable speaker which is accessed via the audio input socket.)

The pitch and duration of the beep can be set by the escape sequence

ESC, '@', f, d, 'A'

where f defines the frequency (a higher value giving a lower note) and d the duration (a higher value giving a longer note). The frequency is approximately:-

$$\frac{1000}{54 + 32*f} \text{ khz}$$

3. CHANGES TO FRONT PANEL AND COS COMMANDS

3.1 Front Panel

The Z command now single steps through an entire EMT, rather than stepping through the EMT code instruction by instruction.

When any of the commands M, J, S, G, P or H is given, the hexadecimal numbers entered may be terminated by one of the following characters only: RETURN; SPACE; CTRL F. If CTRL F is typed at any time, then the command is aborted. All non-hexadecimal characters (other than DELT, RETURN, SPACE and CTRL F) are ignored, including CTRL B and CTRL C. RETURN and SPACE have exactly the same effect.

In the case of the S and P commands, if the first value is greater than the last value, then the message 'Error' is given, and the command is restarted.

The W command toggles the screen width on exit from Front Panel. A message is printed after typing W, indicating what the screen size will be on exit.

On exit from Front Panel by giving the commands K, J, CTRL B or CTRL C, the screen is cleared and it reverts to the entry width, unless the W command has been entered during Front Panel use.

3.2 COS Commands

On switching on, or pressing the reset button, the 380Z will go into 80 character mode, and give the usual COS prompt message. At this point the commands are the same as for earlier versions of COS, plus the additional command W which will toggle the screen width between 40 and 80 mode, and minus the cassette commands L, D and C.

4. EMT DIFFERENCES

4.1 Modified EMTs

OUTC

Additional control characters :-

CTRL G	Sound beeper
CTRL R	Switch Reverse Video on
CTRL T	Switch Reverse Video off
Escape	Begin a command sequence

The EMT automatically takes account of whether in 40 or 80 character mode, and whether to do ordinary scrolling or smooth scrolling. In ordinary mode the screen scrolls up or down. Scroll down occurs on receipt of CTRL K when the cursor is at the top of the screen.

Receipt of ESCAPE initiates a command sequence; the bytes immediately following ESCAPE determine which command sequence, and the values of any parameters. See appendices A and B.

PUTBYT

Still points to PUNV, but this is initialised to be 0FFH.

GETBYT

Still points to RDRV, but this is initialised to be 0FFH.

OUT1

No longer used by the cassette system - available to the User.

GRAFIX

As COS 3.4, and works in 40 character of 80 character mode, but leaves cursor at bottom left of screen.

SCROLL

As COS 3.4, and works in 40 character or 80 character mode, but leaves cursor at bottom left of screen.

CLEAR

As COS 3.4 except that the contents of all registers are now preserved. Also the routine will only work when in 40 character mode.

WAIT

WAIT no longer uses the hardware '1200 Hz' clock and instead there is a

 Copyright January 1982

software simulation. This has the effect of being slightly quicker than the previously slow '1200 Hz' clock.

GETSYN

Deleted.

GETHEX

In COS 3.4, if CTRL B or CTRL C was typed during GETHEX, the input was terminated and either COS was entered or there was a reboot, respectively. In COS 4.0, CTRL B and CTRL C act like any other terminating character, with the corresponding ASCII value left in Register B.

FSTLST

FSTLST behaves the same as before, except that only Carriage Return, space or CTRL F will terminate input. All other non-hexdecimal characters are ignored (except for delete). If CR or space is typed to terminate the two inputs, then the EMT has the same effect as before, but with the CY flag reset. If CTRL F is used to terminate either input, then the sequence is aborted, the registers are unpredictable, and CY is set to indicate the situation.

If the first number is greater than the last number, then the message 'Error' is given, and the sequence restarted. (Previously, after the error message, the routine went to the monitor level.)

SETCAS

Deleted.

OUTF

As COS 3.4 although slower. Only works when in 40 character mode.

INF

As COS 3.4 although slower. Only works when in 40 character mode.

SETTAB

Deleted.

VERIFY

Deleted.

 Copyright January 1982

Notes (i) All the deleted EMTs return directly after a call, with nothing changed.

(ii) The 3 EMTs which only work when in 40 character mode return immediately if in 80 character mode. It is intended that they be phased out, and replaced by the 3 more general EMTs: VTOUT, VTIN and VTCLR.

4.2 Specification of new EMTs

WIDTH (Code 52)

This routine changes the width of the screen or returns the value of the current screen width. On entry, if register A holds 0, then the screen will become a 40 character screen. If A contains 1, then the screen will become an 80 character screen. If a 40 character screen becomes an 80 character screen then the data on the 40 character screen is displayed on the left hand side of the 80 character screen, and the right hand side is blanked. If an 80 character screen becomes a 40 character screen, then each line from the 80 character screen is truncated to become the new 40 character line. The cursor is placed at the bottom left of the screen. If register A contains -1 on entry, then on return A will hold 0 or 1, depending on whether the screen is currently in 40 or 80 mode.

Registers affected : If A = -1 on entry then A,flags
Otherwise, none.

CHGEN (Code 53)

This routine generates a new character dot pattern for characters in the range 128 - 255. On entry, register A contains the character which is to be changed. Register B must contain 0 for normal use, or 1 if the call is to be actioned immediately, rather than waiting for frame blanking. The new dot pattern must be contained in 12 bytes of contiguous memory, the address of the first being held in register DE.

Registers affected : none.

CHREAD (Code 54)

This routine reads the current dot pattern for a character in the range 0 - 255. On entry register A contains the character which is to be read. Register B must contain 0 for normal use, or 1 if the call is to be actioned immediately, rather than waiting for frame blanking. The pattern will be output to 12 bytes of contiguous memory, and on entry DE must contain the address of the first byte.

Registers affected : none.

VTOUT (Code 55)

Outputs a character to the screen at a specified address. The routine works in both 40 and 80 character mode. On entry, register A contains the character to be output, register E contains the attributes, and register HL

contains the (x,y) address from the top left hand corner of the screen. The y value (from 0 to 23) is in H, and the x value (from 0 to 39, or from 0 to 79) is in L. If the (x,y) value is outside these limits, then the call has no effect. Register E should be 0 for a normal character. Otherwise, the attribute bits are :

```
        0 :- Alternate ROM
        1 :- Underline
        2 :- Dim character
        3 :- Reverse video
4 to 7 :- Ignored
```

Registers affected : none.

VTIN (Code 56)

Reads a character from the screen at a specified address. The routine works in both 40 and 80 character mode. On entry register HL contains the (x,y) address from the top left hand corner of the screen. The y value (from 0 to 23) is in H, and the x value (from 0 to 39, or from 0 to 79) is in L. If the (x,y) value is outside these limits, then registers A and E are contained zero on return. On exit register A contains the character at that address, and E contains the attributes. See VTOUT for a list of the attribute bits. Bits 4 to 7 of E are returned as zero.

Registers affected : A,E,flags.

VTCLR (Code 57)

This routine clears a specified area of the screen. The routine works in both 40 and 80 character mode. On entry register B contains the top line of the area , register C contains the bottom line, register D contains the leftmost column, and register E contains the rightmost column.

0 <= B <= C <= 23.
0 <= D <= E <= 39 or 79 (depending on whether 40 or 80 character mode).

If any of the vaues of B, C, D or E do not conform to these limits, then the call has no effect. The character positions that have been cleared have had a graphics blank character (ASCII 80 hex) written to them.

Registers affected : none.

WINDOW (Code 58)

This routine changes the allowable cursor area to a window, with scrolling happening within this window. The routine works in both 40 and 80 character mode. On entry register B contains the top line of the window, register C contains the bottom line, register D contains the leftmost column, and register E contains the rightmost column.

0 <= B <= C <= 23.
0 <= D <= E <= 39 or 79 (depending on whether 40 or 80 character mode).

If any of the values of B, C, D or E do not conform to these limits,then

the call has no effect. After the new window has been defined, the cursor is placed at the bottom left hand corner of the new window, no matter where the cursor was before the call, or what the previous window was.

Registers affected : none.

CURPOS (Code 59)

This routine returns the address of the cursor position, in (x,y) form. On return, H contains the y value, and L contains the x value.

Registers affected : HL.

VTLINE (Code 60)

This routine outputs a line or part of a line in 40 or 80 character mode. HL contains the start address on the screen, in x,y form, and does not have to be the start of a line. DE contains the start address of the string which is to be output. Register C contains the number of characters which are to be output, but output will stop if the end of the screen line is reached before then.

Registers affected : none.

5. ARCHITECTURE DIFFERENCES

5.1 Memory maps

The additional features in COS 4.0 have necessitated the use of more ROM space. The additional ROM is mapped as 1K bytes (from 0F600H to 0F9FFH) in the map which is used in normal operation (page 1), and an additional 512 bytes which is only available via the map which is used during initialisation (page 0).

COS changes memory maps when it requires to execute code on Page 0. Interrupts are inhibited during this process.

The facility for adding an option ROM is not supported by COS 4.0.

Research Machines reserves the right to alter the memory map in future versions of COS.

5.2 Addresses used by COS

The following entrypoints to COS have been preserved. It should however be noted that these entrypoints are not identical on the LINK 480Z.

E010 - BREAK
E013 - Entry to Front Panel

The following locations in the COS scratchpad have been preserved:-

BASE + 0012 KBDTLV - The KBDTL vector
BASE + 0015 KBDWFV - The KBDWF vector

BASE + 0018 VTV - Console out vector
BASE + 001B KBDV - Console in vector
BASE + 001E PUNV - Tape out vector (but now contains FF)
BASE + 0021 RDRV - Tape in vector (but now contains FF)
BASE + 0024 LPV - Line printer out vector → FF18 0
BASE + 0027 OUT1 - Output vector 1 EF0C 1
BASE + 002A OUT1 - Output vector 2 EF15 2
BASE + 002D IN1 - Input vector 1 EEFB 3
BASE + 0030 IN2 - Input vector 2 EF4A 4
BASE + 0033 IN3 - Input vector 3 EF54 5
BASE + 0036 TRAPX - Trap extension vector EF5A 6
BASE + 0039 MONX - Monitor extension vector

BASE + 003E UMASK - The User I/O Port mask

BASE + 006E INITV - Disc initialisation vector
BASE + 0071 RDSECV - Disc read sector vector
BASE + 0074 WRSECV - Disc write sector vector
BASE + 0077 WRCHKV - Disc write sector and check vector.

Other locations have changed function. For example VTPTR (the position of the cursor) has remained at BASE + 0004, but is now in (x,y) form, instead of a memory map value. Other locations may have remained the same, but cannot be assumed to do so in further versions of COS.

X → FF04
Y → FF05

Accessing COS workspace directly is not recommended, and EMTs should be used where possible. If it is essential to access the workspace it should be regarded as being relative to the address returned by EMT BASE.

5.3 Screen access methods

In 40-character mode the screen access is compatible with COS 3.4, and memory mapped access to the VDU RAM is possible.

In 80-character mode the screen is mapped differently and cannot all be accessed at once. PORT 1 (0FBFEH) bits 0 to 3 are set to bits 1 to 4 of the Y-address of the position to be accessed.

The address used is then

0F000H + X-address + (128 * Y bit 0)

It is strongly recommended that all screen access is performed using EMTs rather than using direct access to the VDU memory, since this will facilitate transfer of programs to other systems (e.g. the LINK 480Z and future 380Z configurations).

APPENDIX A. Escape Sequences

This section defines all the available escape sequences for reference purposes. Examples of the most common ones are given in section 2. All escape sequences begin with the ESCAPE character (ASCII 27). The first character after the ESCAPE is called the Sequence Introducer (S.I.), and it determines the number of characters after the S.I. that are assumed to be part of the escape sequence. If the Sequence Introducer is not recognised, then the escape sequence is immediately terminated, and any futher characters in the sequence are displayed on the screen as normal. Illegal characters after a legal S.I. will invalidate the sequence, but will not affect the number of characters which are grabbed. Any escape sequence which is not recognised or contains one or more illegal characters is not executed.

Note that only charactars output via EMT OUTC and EMT MSG are considered as candidates for escape sequences. Moreover, if screen output is switched off by CTRL O, escape sequences will have no effect.

In the following list of escape sequences, CP is a Control Parameter, chosen from the given list, and SW is a switch value, chosen from the given list.

1) Sequence Introducer is '!' (ASCII 33)

ESC, '!', c

The symbol corresponding to code c is output directly to the screen via OUTC, but without cursor control codes being interpreted.

2) Sequence Introducer is '<' (ASCII 60)

ESC, '<', c, b1, b2, b3, b4, b5, b6, b7, b8, b9, b10, b11, b12

Define the dot pattern in the Writable Character Store for character c (in the range ASCII 128 to 255); b1 to b12 are the dot pattern.

3) Sequence Introducer is '=' (ASCII 61)

ESC, '=', SW, CP

SW = '0' or '1' to switch the feature off or on.

CP is 'A'	Switch off or on the alternate ROM attribute
CP is 'B'	Switch off or on the underline attribute.
CP is 'C'	Switch off or on the dim attribute.
CP is 'D'	Switch off or on the reverse video attribute.
CP is 'E'	Switch off or on the 8 bit keyboard.
CP is 'F'	Switch off or on the CTRL F action.
CP is 'G'	Switch off or on the CTRL @ and CTRL A action.
CP is 'H'	Switch off or on smooth scrolling.
CP is 'I'	Switch off or on the text in HRG output.
CP is 'J'	Switch to 40 or 80 character mode. ('0' = 40, '1' = 80)
CP is 'K'	Switch off or on the automatic dimming of ASCII characters 128 to 191 in 80 character mode.

 Copyright January 1982

4) Sequence Introducer is '>' (ASCII 62)

ESC, '>', CP

CP is 'A' Clear current attributes, clear smooth scroll, set to 80 character mode, full screen window, put cursor at bottom left, and initialise the Writable Character Store to be the usual graphics set.

CP is 'B' Initialise the Writable Character Store to be usual graphics set.

CP is 'C' Clear the current attributes.

5) Sequence Introducer is '?' (ASCII 63)

ESC, '?', XL, XU, YL, YU, CP

CP is 'A' The 4 bytes defines a scrolling window, in the order x lower, x upper, y lower, y upper.

0 <= XL <= XU <= 39 or 79.
0 <= YL <= YU <= 23.

CP is 'B' The 4 bytes define a rectangular area to be cleared. Parameters as above.

6) Sequence Introducer is '@' (ASCII 64)

ESC, '@', F, D, CP

CP is 'A' The 2 bytes F and D define the frequency and duration of the beeper (which is sounded by outputting CTRL G).

$$\text{frequency} = \frac{1000}{54 + 32*F} \text{ khz.}$$

 Copyright January 1982

APPENDIX B. Effect of control characters on the screen

	CTRL	Committed	New
0	@	RESERVED (NULL)	
1	A		
2	B		
3	C		
4	D	Resume output	
5	E		
6	F		
7	G		Beep
8	H	Cursor left	
9	I	Tab	
10	J	Cursor Down	
11	K	Cursor up	
12	L	Clear screen (dursor BL)	
13	M	CR + LF	
14	N	CR	
15	O	Suppress output	
16	P		
17	Q	Clear auto-paging	
18	R		Reverse video on
19	S	Set auto-paging	
20	T		Reverse video off
21	U	Blinking cursor on	
22	V	Initiate XY addressing	
23	W	Blinking cursor off	
24	X	Cursor right	
25	Y	Clear to end of line	
26	Z		
27	[		Begin an escape sequence
28	\		
29	]	Cursor home (TL)	
30	↑	Clear to end of screen	
31	←	Cursor home and clear screen	
127	DELT	Destructive Backspace	

 Copyright January 1982

APPENDIX C. COS 4.0 Quick Reference Guide

SUMMARY OF TRAP CALLS

Mnemonic	Code	Hex	Function
ERROR	-1	-1	Print Error, return to Command Level
CONTC	0	0	Return to Command Level
OUTC	1	1	Output byte in reg A to screen
KBDC	2	2	Read keyboard, trap CTRL C
-PUTBYT	3	3	Output byte in reg A to tape
-GETBYT	4	4	Read byte from tape into reg A
LPOUT	5	5	Output byte in reg A to selected printer
OUT1	6	6	EMT via transfer vector (spare)
OUT2	7	7	As OUT1
IN1	8	8	EMT via transfer vector (spare)
IN2	9	9	As IN1
IN3	10	A	As IN1
OPNWT	11	B	Open screen in frame blanking
CLOSE	12	C	Close screen
-*GRAFIX	13	D	Clear screen, 4 line scroller
-*SCROLL	14	E	Full screen scroller
-*CLEAR	15	F	Clear selected screen area
WAIT	16	10	Wait B 833 usec intervals
-GETSYN	17	11	Get sync char in reg A from tape
UPDATE	18	12	Copy MASK to PORT0
GETHEX	19	13	Get hex number from kbd in reg HL
DEOUT	20	14	Put reg DE to (HL) in ASCII
BYTEO	21	15	Put reg A to (HL) in ASCII
*OUTC	22	16	As EMT 1 (not via transfer vector)
MSG	23	17	Output message at (HL) to screen
CHAN	24	18	Execute EMT given in reg C
INIT	25	19	Initialise disk unit
RDSEC	26	1A	Read sector
WRSEC	27	1B	Write sector
WRCHK	28	1C	Write and check sector
KBDIN	29	1D	Read and clear keyboard
KBDTC	30	1E	Test keyboard, return character
KBDTL	31	1F	Test keyboard, return -1 (true)
KBDTF	32	20	Test keyboard for CTRL F
KBDW	33	21	Wait for character from keyboard
KBDWF	34	22	As KBDW, trap CTRL F
SAVE	35	23	Save regs HL, DE and BC
SAVEA	36	24	As SAVE, also save AF
FSTLST	37	25	Access COS routine
GETJP	38	26	Return Ath entry from table at (HL)
BASE	39	27	Return scratchpad addr in reg DE
-SETCAS	40	28	Set tape speed from reg A
SETLST	41	29	Set printer from regs A and E
OUTNC	42	2A	Scroller output without clearing page key
-*OUTF	43	2B	Output to screen at (HL) from reg A
-*INF	44	2C	Input from screen at (HL) to reg A
S4KTL	47	2F	As KBDTL, but from SIO-4
S4KIN	48	30	Read SIO-4 into reg A
BOOT	49	31	Perform a cold bootstrap
LPSTAT	50	32	Check that printer is ready
VERSN	51	33	Return version number of COS

 Copyright January 1982

COS 4.0 and not COS 3.4 :-

WIDTH	52	34	Set, or interrogate, screen width
CHGEN	53	35	Generate writable character dot pattern
CHREAD	54	36	Read character dot pattern
VTOUT	55	37	Output character A, attributes E to screen
VTIN	56	38	Read character from screen to A, attributes to E
VTCLR	57	39	Clear screen area
WINDOW	58	3A	Set scrolling window
CURPOS	59	3B	Read cursor position
VTLINE	60	3C	Output line to screen

* These EMTs may not be supported in future versions of COS
\- These COS 3.4 EMTs are not fully supported by COS 4.0

SUMMARY OF MONITOR COMMANDS

B	Bootstrap CP/M from disc and execute
X	Bootstrap CP/M from disc drive B
J	Jump to address and execute program
O	Select printer option
M	Enable High Resolution Graphics board as memory
W	Change screen width
CTRL C	Return to Command Level
CTRL F	Enter Front Panel mode
CTRL T	Enter Typewriter mode

SUMMARY OF FRONT PANEL COMMANDS

RETURN	Increment memory pointer
-	Decrement memory pointer
LINE FEED	Advance memory pointer by 8
/	Move memory pointer back by 8
CTRL L	Advance memory pointer by 32
CTRL O	Move memory pointer back by 32
M	Set memory pointer
I	Set memory pointer from memory contents (absolute)
R	Set memory pointer from memory contents (relative)
U	Set memory pointer from PC contents
.	Increment register pointer
X	Switch register display to alternate set
V	Update PC from memory pointer
G	Get first occurrence of specified pattern of bytes
N	Find next occurrence of pattern
H	Hexadecimal calculator
P	Fill and test memory between limits
S	Shift memory content
O	Set printer option
W	Change screen width (on exit)
J	Jump to address and start execution
K	Continue program execution from PC
Z	Single step

 Copyright January 1982

<	Increment I/O port pointer
>	Decrement I/O port pointer
,	Write byte to I/O port
CTRL B	Return to monitor command level
CTRL C	Go to either command level or to CP/M if booted

 Copyright January 1982